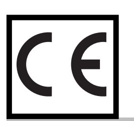
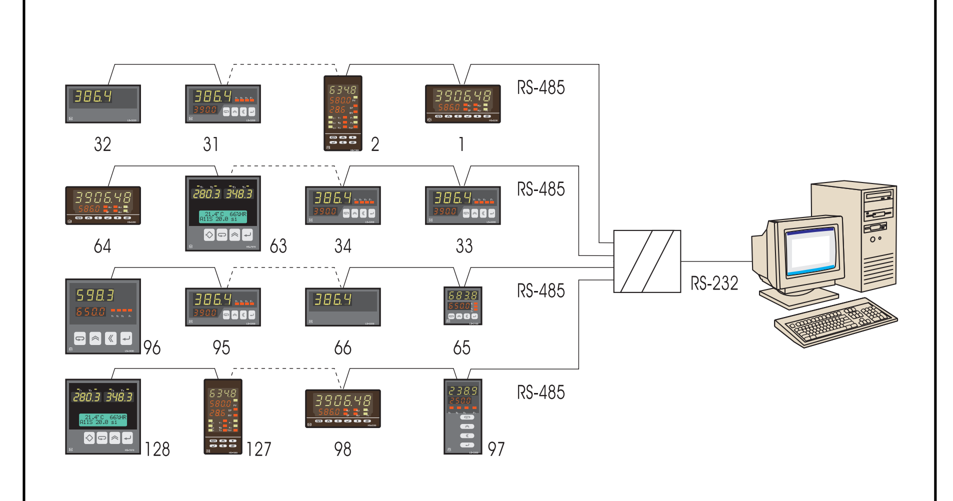
COMUNICACIONES RS-485 MODBUS



INSTRUCCIONES AVANZADAS

SERIES DAS- 8000 LS-3000 MS-5000 HS-7000



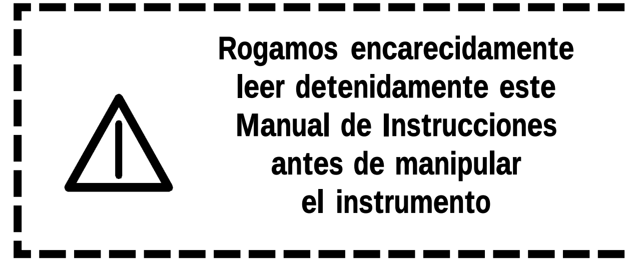




INDICE GENERAL

RECOMENDACIONES BÁSICAS

Este Manual está dirigido expresamente al responsable de instrumentación que tenga a su cargo la configuración y puesta a punto de estos aparatos para su óptima aplicación.



NOTA: Estos instrumentos son expedidos de fábrica con un nivel de protección que permite el acceso a los parámetros de visualización, protegiendo el resto de submenús con claves de acceso (passwords) con el fin de evitar que por desconocimiento de su utilización puedan alterarse los datos de programación o configuración guardados en su memoria.

Estos **MANUALES DE INSTRUCCIONES** son ampliados continuamente por nuestro departamento de ediciones, generando nuevas versiones en formato PDF que pueden descargarse libremente de nuestra web:

www.desin.com

MUY IMPORTANTE

Esta sección es aplicable sólo para modelos con salida de comunicación RS-232 o/y RS-485. Protocolo MODBUS.

DESCRIPCIÓN DE LA RED DE COMUNICACIONES

El objetivo de una red de comunicaciones no es más que el de unir una serie de *dispositivos* para que intercambien información entre ellos. Toda red se compone de un medio físico (ethernet, token ring, RS232, RS485, ...) por el que circula la información, y de un protocolo de comunicaciones (TCP/IP, Modbus, ...) que no es más que el lenguaje que permite un entendimiento entre ellos. Por tanto, todos los dispositivos conectados en una red deberán soportar tanto el medio físico como el protocolo especificado.

El presente documento se basa en una red RS485 y protocolo MODBUS-RTU.

MEDIO FÍSICO

La red RS485, usada en entornos industriales, supone un eslavón superior, respecto a su predecesor (RS232), al permitir largas distancias entre dispositivos (hasta los 1200 metros entre extremos), y la conexión de hasta 32 dispositivos (pudiéndose ampliar con repetidores hasta los 255). Debido a que en este tipo de redes, el dispositivo principal suele ser un PC o PLC y que estos disponen de puertos de comunicaciones RS232, la conversión de la señal se realizará con lo que se denomina *Convertidor RS232/485*.

Este tipo de red permite realizar cableados a 2 ó 4 hilos, para intercambio de información Half o Full duplex respectivamente. Este documento reflejará, mediante dibujos explicativos, las diferentes conexiones con los dispositivos antes mencionados.

El cableado se realizará a 2 hilos. Si se dispone de convertidores o puertos serie con salida RS485 a 4 hilos, para pasarlo a 2 hilos bastará con efectuar un puente entre bornes con el mismo signo; TX positivo con RX positivo, y TX negativo con RX negativo.

PROTOCOLO

El protocolo de comunicaciones Modbus es un lenguaje de red tipo MAESTRO - ESCLAVO (MASTER - SLAVE), en el cual la comunicación sigue el principio de Pregunta/Respuesta.

En su modalidad RTU, se caracteriza porque, cada byte, 8 bits, del mensaje contiene dos caracteres hexadecimales de 4 bits. La ventaja principal de este modo es que su mayor densidad de caracteres permite una mejor productividad de información que el modo ASCII para la misma velocidad.

El *Maestro* (único en toda la red), genera los mensajes de petición, mientras que los *Esclavos* proporcionan la respuesta a esas peticiones. Hay que tener presente que sólo el Maestro puede iniciar el intercambio de mensajes, y que sólo debe depositar en la red un mensaje en espera de una respuesta; de otro modo, se produciría un «choque» de mensajes dando lugar a errores. De igual forma, dos maestros comunicando a la vez, provocarían conflictos en las comunicaciones al depositar varios mensajes simultáneamente en la red.

Este protocolo permite direccionar hasta 255 dispositivos esclavos bajo la norma RS485 en estrella. *Direccionar* se define como la asignación de un número, único en toda la red, que identifica al dispositivo y que permitirá que éste reconozca los mensajes que le vayan dirigidos. Ciertas características del protocolo Modbus son fijas, como el formato del mensaje, manejo de los errores de comunicación, condiciones de excepción y las funciones a realizar.

Otras características, aunque configurables por el usuario, vienen condicionadas por el propio dispositivo al presentar diferentes formas de realizar las operaciones. Estas características incluyen la elección del medio de transmisión, el formato de comunicaciones y la velocidad de comunicaciones.

El presente manual se basa en los dispositivos fabricados por DESIN Instruments, tales como AC-1000, DAS-8000, LS-3000, MS-5000 o HS-7000.

Para todos ellos la comunicación se realiza en el siguiente formato:

Velocidad: Seleccionable entre 9600, 19200 ó 38400 baudios.
Paridad: Sin paridad (Nula).
Bits de datos: 8 bits.
Bits de parada: 1 bit.

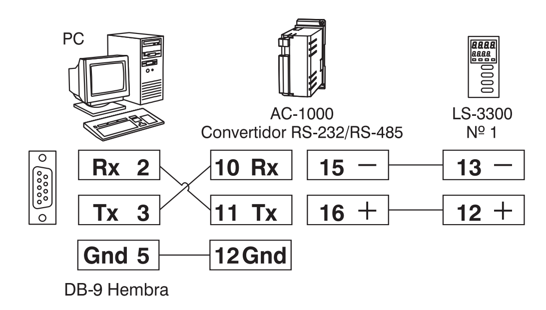
La presencia de un PC (o PLC) en la red impone que éste sea el Maestro y el resto de los dispositivos, los Esclavos. Este PC deberá configurarse para establecer qué puerto serie (COM1,2,3,4,...) utilizará para comunicarse con los esclavos y los parámetros de velocidad y paridad antes descritos que deberán ser idénticos a todos los miembros de la red.

De igual forma, la presencia del convertidor RS232/485, AC-1000, debido a que forma parte de uno de los dispositivos de la red modbus, deberá llevar asignado un número que lo diferencie del resto de dispositivos. De fábrica se entregará con la dirección 255, por lo que en la red podrán conectarse hasta 254 dispositivos adicionales.

CONEXIONADO

Seguramente, el primer dispositivo que se deberá instalar en la red será un convertidor RS232/485 para adaptar la salida serie del PC (PLC) a la línea de comunicaciones. *Para asegurar la fiabilidad de las comunicaciones, es totalmente indispensable utilizar un convertidor que disponga de* ***aislamiento galvánico*** *mediante optoacopladores entre la entrada RS-232 y la salida RS-485*. El **AC-1000** es un convertidor de comunicaciones inteligente de RS-232/485 con aislamiento galvánico para aplicaciones industriales que utilicen el protocolo Modbus-RTU, que puede montarse en raíl DIN.

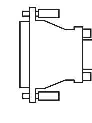
Conexión como Convertidor RS232 (puerto serie de PC o PLC) a RS485 Modbus



NOTAS:

Si el conector del ordenador es un DB-25 Macho, se utilizará un adaptador DB9 Macho a DB25 Hembra, o bien se sustituirá el conector DB9 Hembra por un conector DB25 Hembra según la tabla adjunta.

Conector 9 pines	Conector 25 pines
Rx 2	Rx 3
Tx 3	Tx 2
Gnd 5	Gnd 7



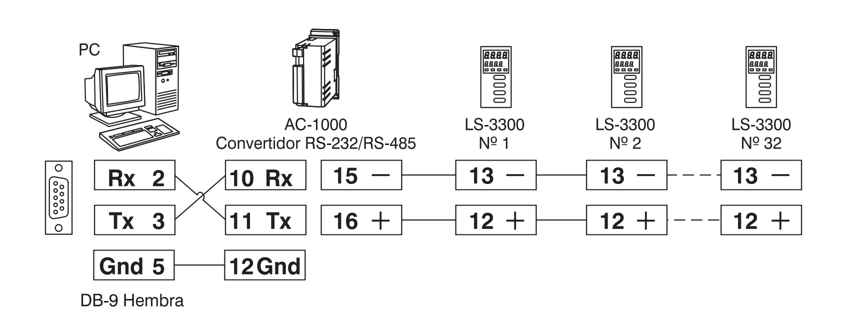
La distancia máxima que puede haber entre el convertidor de RS-232 a RS-485 y el Ordenador es de 15 metros como máximo.

Los siguientes apartados describen los diferentes tipos de conexiones que pueden realizarse utilizando un AC-1000. Para una información más exhaustiva de este dispositivo, leer el *Manual de Instrucciones Avanzado del AC-1000.*

Conexión en RS-485 hasta 32 dispositivos tipo LS-3000, MS-5000, HS-7000 o DAS-8000 y una distancia máxima de 1200 metros

Esta suele ser la conexión estándar. Un solo convertidor que conecta hasta 32 dispositivos con un PC o PLC (Las direcciones de estos 32 sispositivos podrán ser arbitrarias pero dentro del intervalo de 1 a 254; el dispositivo 255 será el convertidor).

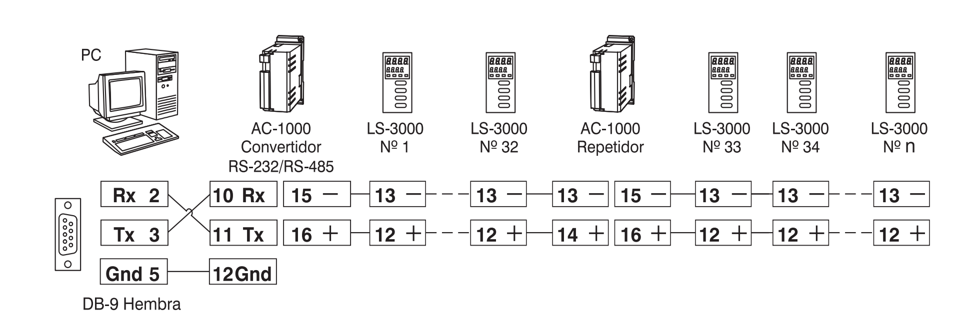
Nota: La figura adjunta presenta un ejemplo de conexionado sólo con LS-3000 y estructura de bus. Consultar el conexionado RS485 para cada dispositivo en esta guia.



Conexión de un AC-1000 funcionando como Repetidor serie en RS-485 para unir más de 32 dispositivos o distancias superiores a los 1200 metros

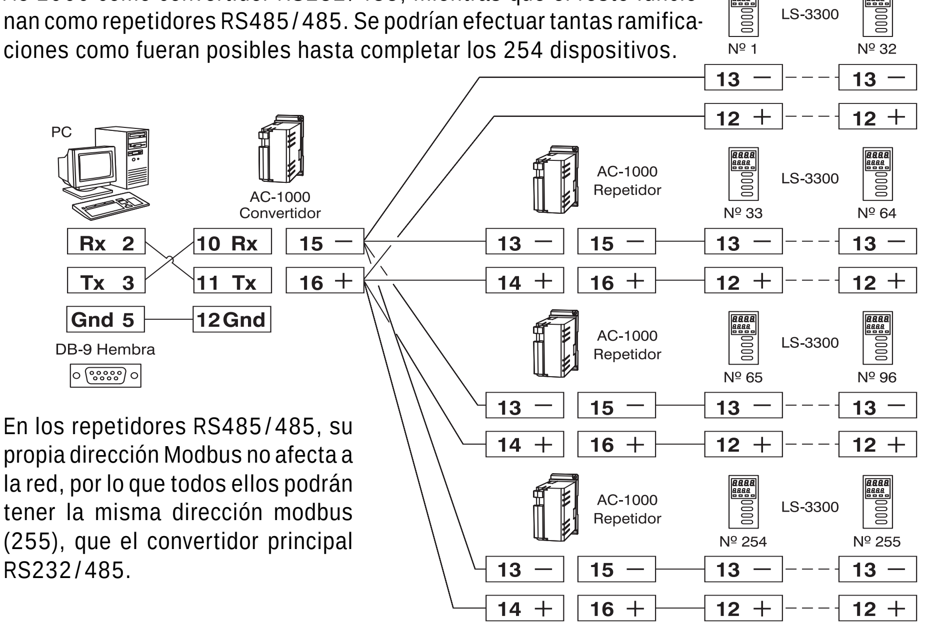
El **AC-1000** configurado como repetidor, permite formar redes hasta cubrir los 254 dispositivos de conexión, con una distancia total de líneas de bus de varios kilómetros, extendiendo las líneas de comunicación RS-485 en arquitecturas de tipo bus (figura adjunta), estrella o árbol, sin perder nivel de señal y mejorando el aislamiento entre puntos de la red.

Hay que tener presente que, el AC-1000, como dispositivo MODBUS-RTU, debe tener asignada una dirección específica y única en la red. Este dispositivo se entregará de fábrica con la dirección 255, permitiendo al usuario la utilización de las direcciones bajas de direccionamiento.



Conexión de un AC-1000 funcionando como repetidor, estructura de árbol/estrella, para conectar hasta 254 dispositivos tipo LS-3000, MS-5000, HS-7000 o DAS-8000

Por estructura del emplazamiento, el AC-1000 permite realizar conexiones en estructura de estrella, permitiendo varias ramificaciones hacia las diferentes áreas de la empresa. Observar la existencia de un primer AC-1000 como convertidor RS232/485, mientras que el resto funcionan como repetidores RS485/485. Se podrían efectuar tantas ramificaciones como fueran posibles hasta completar los 254 dispositivos.



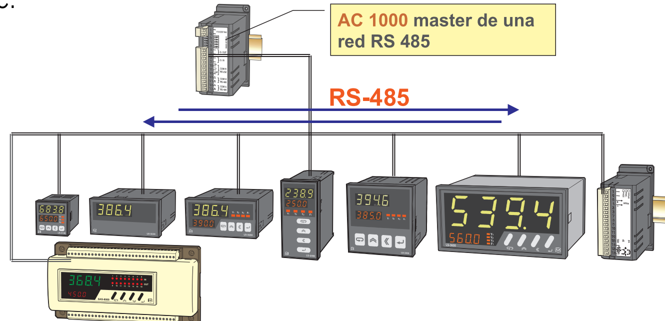
En los repetidores RS485/485, su propia dirección Modbus no afecta a la red, por lo que todos ellos podrán tener la misma dirección modbus (255), que el convertidor principal RS232/485.

Conexión de un AC-1000/LM como Master de Comunicaciones MODBUS

El adaptador versión **AC-1000/LM** permite intercambiar datos en campo a nivel horizontal entre dispositivos de control, sin necesidad de ser comandados por PC o PLC.

Para realizarlo, dispone de un bloque LINKER de comandamiento que posibilita 40 pasos de programación y 20 registros, permitiendo leer datos directamente de las posiciones de memoria de un dispositivo y escribirlos sobre la memoria de otro dispositivo.

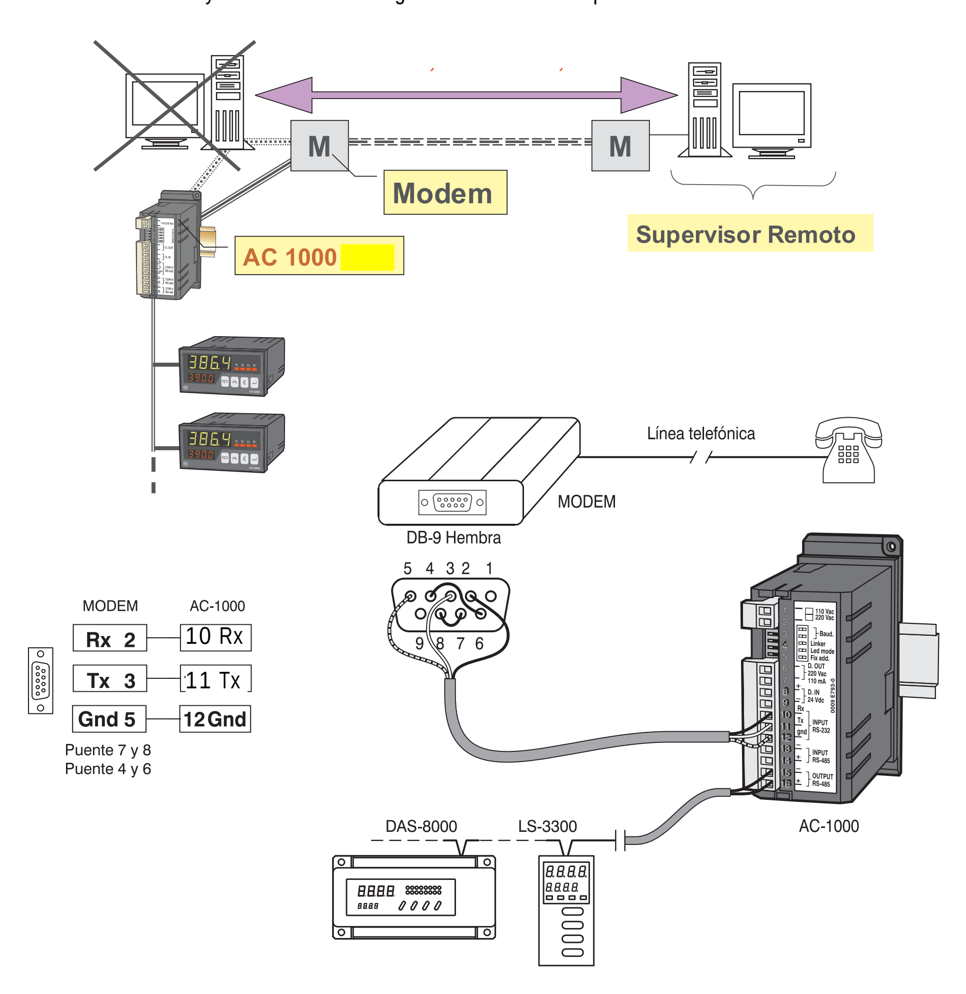
El **AC-1000/LM** actúa siempre de forma transparente para el resto de información que se transmita por la línea de comunicación RS-485 Modbus, permitiendo su funcionamiento en paralelo con un PC.



Conexión de un AC-1000 controlado por un Modem Telefónico

El convertidor **AC-1000** puede conectarse a un Módem estándar tipo Hayes, sin necesidad de PC, para aplicaciones en las que se precise supervisar una red de instrumentos de control de forma remota. En esta función, el **AC-1000** recibe, a través del módem, los protocolos MODBUS que deberá transmitir a la línea RS485. La salida serie del modem deberá conectarse a la entrada RS232 del AC-1000, y el modem deberá programarse para que se conecte o desconecte a recibir o cortar una llamada telefónica.

Al otro lado de la línea telefónica, un PC conectado a un modem se encargará de efectuar la llamada. Una vez establecida la comunicación entre modems, podrá acceder a todos los datos de la red Modbus y comandar o configurar hasta 254 dispositivos.



Importante

Más información sobre estas dos últimas conexiones en el Manual Avanzado del AC-1000

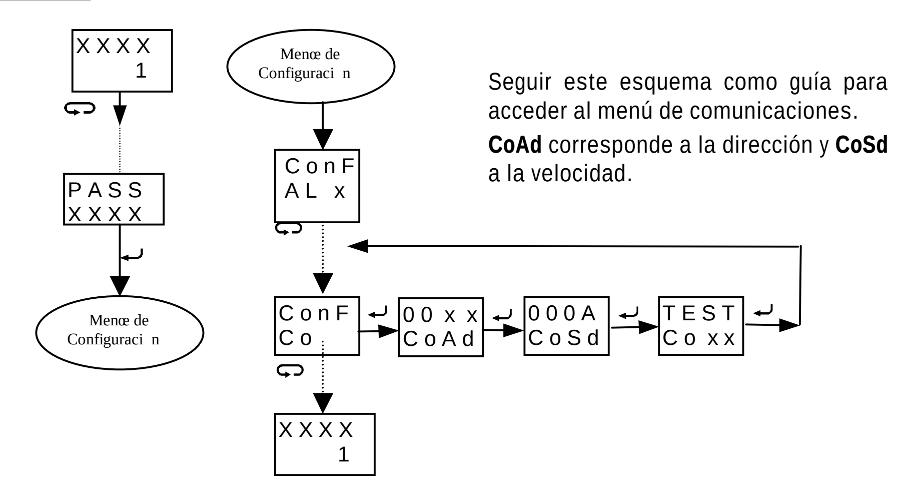
DIRECCIONAMIENTO Y VELOCIDAD DE LOS DISPOSITIVOS

Como se comentó anteriormente, cada dispositivo esclavo estará identificado en la red por un único número establecido entre 1 y 255. Por contra, el parámetro que especifica la velocidad de comunicaciones deberá ser común a todos los dispositivos. Estos dos valores deberán introducirse por medio del teclado del dispositivo (si el modelo lo incorpora), o mediante un software de configuración, **LoopWin,** funcionando en el PC.

Si se configura por teclado, la dirección que se establecerá para cada módulo será un número de dos dígitos en hexadecimal. La velocidad, 9600, 19200 y 38400, está codificada con los valores 0, 1 y 2 respectivamente.

La configuración por software LoopWin se describe en el manual avanzado de cada dispositivo. Acto seguido se describirán los pasos para acceder a los parámetros de comunicaciones de cada dispositivo con el fin de mostrar cómo direccionarlos antes de conectarlos a la red.

DAS-8000



Seguir este esquema como guía para acceder al menú de comunicaciones.

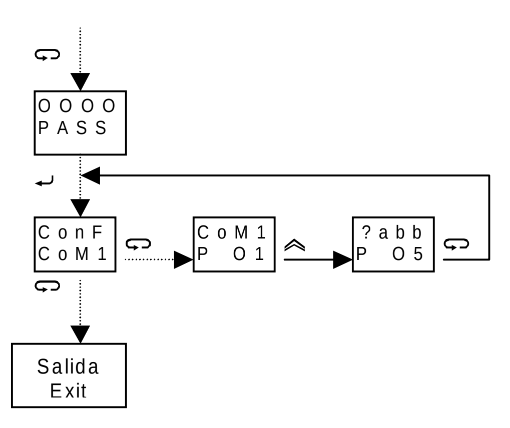
CoAd corresponde a la dirección y **CoSd** a la velocidad.

LS-3000

En el parámetro «**P 05**» se especifica la velocidad y numeración del dispositivo.

«**a**» En este parámetro se indicará la velocidad. 0(9600), 1(19200), 2(38400)

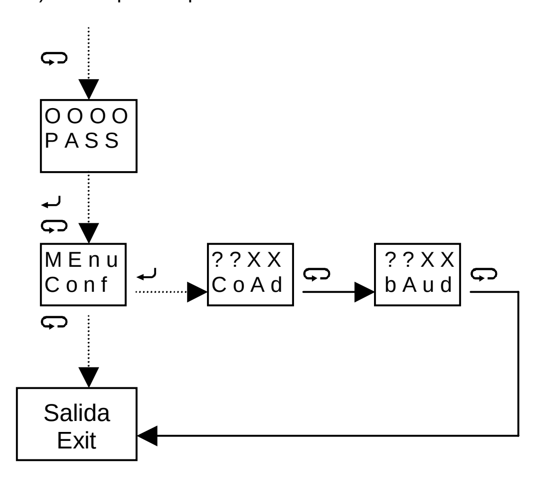
«**bb**» Este parámetro contiene la dirección del dispositivo en hexadecimal.



MS-5000

«**CoAd**» contiene la dirección del dispositivo en los dígitos XX. Esta dirección está indicada en hexadecimal. Para modificar el valor, pulsar la tecla ENTER para habilitar la edición, con la tecla de incremento especificar el nuevo valor, y volver a pulsar ENTER para validar.

«**bAud**» especifica la velocidad del puerto de comunicaciones (un dígito por puerto): 00 (9600), 10 (19200), 20 (38400). Cada puerto puede tener una velocidad diferente.

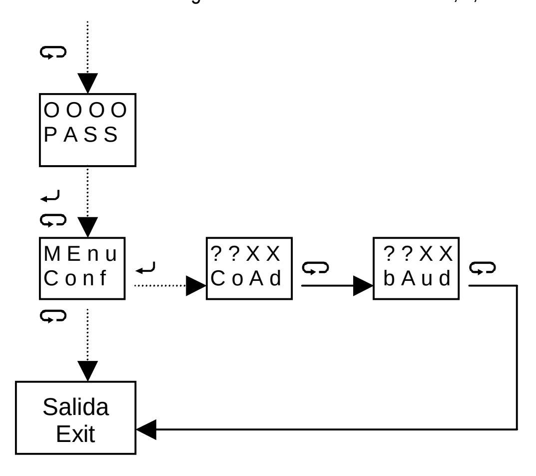


HS-7000

«**CoAd**» contiene la dirección del dispositivo en los dígitos XX. Esta dirección está indicada en hexadecimal. Para modificar el valor, pulsar la tecla ENTER para habilitar la edición, con la tecla de incremento especificar el nuevo valor, y volver a pulsar ENTER para validar.

«**bAud**» especifica la velocidad de los dos puertos de comunicaciones (un dígito por puerto): 0(9600), 1(19200), 2(38400). Cada puerto puede tener una velocidad diferente.

Los puertos vienen identificados de la siguiente forma: ??XX ----> ?,?,COM1,COM2

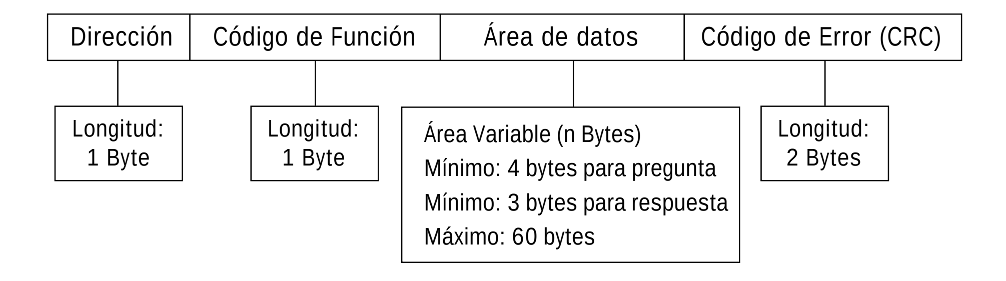


DESCRIPCIÓN DEL PROTOCOLO DE COMUNICACIONES

Los protocolos de comunicación son los diferentes caracteres y códigos que utiliza un dispositivo inteligente para comunicarse con otros. Si ambos conocen este lenguaje podrán intercambiarse información (un símil con los protocolos sería el idioma que hablan y entienden los habitantes de un país, pero que impediría comunicarse con los habitantes de otros países).

ESTRUCTURA DE LOS MENSAJES

Los mensajes que se van a intercambiar entre el ordenador o sistema inteligente y el resto de dispositivos conectados a la línea de comunicaciones RS485, tendrán siempre el mismo formato. Estos mensajes son binarios, y no utilizan ningún carácter que identifique el inicio y el final del mensaje.



Un mensaje MODBUS-RTU está estructurado de la forma siguiente:

DIRECCIÓN: (1 byte). Número de Esclavo con el que se comunicará. Es un valor comprendido entre 1 y 255 (1 a FF en hexadecimal).

CÓDIGO DE FUNCIÓN: (1 byte). El campo Código Función le indica al dispositivo direccionado, la función a realizar y sobre qué área de la memoria (de lectura o lectura/escritura) debe actuar.

Los dispositivos fabricados por DESIN Instruments, soportan los siguientes códigos:

Código de Función	Descripción
3	**Lectura** de N palabras del área de **lectura/escritura.**
4	**Lectura** de N palabras del área de **lectura.**
6	**Escritura** de 1 palabra en el área de **escritura** (nuevo).
16	**Escritura** de N palabras en el área de **lectura/escritura.**

ÁREA DE DATOS: En dependencia del Código Función anterior, tendrá una dimensión diferente, siendo de un máximo de 60 bytes, y un mínimo de 3 bytes para un mensaje de respuesta o un mínimo de 4 bytes para uno de pregunta.

Área de datos para un mensaje de **Pregunta**.

Código de Función	Área de Datos de **PREGUNTA**
3	**2 bytes (Hi-Lo):** Dirección de la primera palabra a **Leer** **2 bytes (Hi-Lo):** Cantidad de palabras a leer (máximo 29 palabras)
4	**2 bytes (Hi-Lo):** Dirección de la primera palabra a **Leer** **2 bytes (Hi-Lo):** Cantidad de palabras a leer (máximo 29 palabras)
6	**2 bytes (Hi-Lo):** Dirección de la palabra sobre la cual escribir. **2 bytes (Hi-Lo):** Valor a escribir
16	**2 bytes (Hi-Lo):** Dirección de la primera palabra a **Escribir** **2 bytes (Hi-Lo):** Cantidad de palabras a **Escribir** **1 byte:** Número de palabras por 2 **n bytes (Hi-Lo):** Valores a escribir para cada palabra

Área de datos para un mensaje de **Respuesta**.

Código de Función	Área de Datos de **RESPUESTA**
3	**1 byte:** Longitud en bytes del mensaje **n bytes (Hi-Lo):** Lecturas de las palabras (2 bytes por palabra)
4	**1 byte:** Longitud en bytes del mensaje **n bytes (Hi-Lo):** Lecturas de las palabras (2 bytes por palabra)
6	**2 bytes (Hi-Lo):** Dirección de la palabra modificada. **2 bytes (Hi-Lo):** Valor escrito en la memoria.
16	**2 bytes (Hi-Lo):** Dirección de la primera palabra escrita **2 bytes (Hi-Lo):** Cantidad de palabras escritas

CÓDIGO DE ERROR (CRC) (2 bytes). El campo Código de Error usa la secuencia de chequeo de error **CRC-16**. Utiliza para ello los 2 últimos bytes del mensaje.

Antes de la transmisión de cada mensaje, se calcula y se añade al mensaje una secuencia de chequeo de error, llamada **CRC-16** (**C**yclic **R**edundancy **C**heck). El receptor recalcula el **CRC-16** con el mensaje recibido y lo compara con el **CRC-16** transmitido para comprobar su buena recepción. De ser diferentes, el Esclavo retornará un código de error.

Si el error se produce en las comunicaciones (dispositivo desconocido…), no se obtendrá un mensaje de respuesta. Se recomienda, por tanto, programar el Maestro de forma que, si no hay respuesta en un tiempo razonable, considere que se ha producido un error de comunicaciones. El periodo de este tiempo depende de la velocidad en baudios, la longitud del mensaje y el tiempo de ciclo del Esclavo. Una vez determinado este tiempo (TIME-OUT), el Maestro puede ser programado para retransmitir el mensaje automáticamente.

NOTAS:

a) Para añadir los 2 bytes del CRC al mensaje, primero se añadirá el byte bajo (low), y después el byte alto (high). *Observar que es inverso a los bytes de datos*.

b) Si se desea comprobar si el **CRC-16** es correcto, debe calcularse el **CRC** del mensaje completo, incluido el **CRC**. Si este cálculo es 0, el mensaje es correcto.

Ejemplos en **C** y **BASIC** para el cálculo del **CRC-16**

Función para generar CRC-16 en C

La función tiene dos argumentos:

unsigned char* pMsg : Un puntero al mensaje.
unsigned short sLen : La cantidad de bytes que contiene el mensaje.

```
unsigned short CRC16( unsigned char* pMsg, unsigned short sLen)
{
	unsigned bit, crc, flag;

	crc = 0xFFFF;

	while( sLen > 0 ) {
		crc ^= *(( unsigned char* )pMsg );
		for( bit = 0; bit < 8; bit++ ) {
			flag = crc & 1;
			crc >>= 1;
			if( flag == 1 ) crc ^= 0xA001;
		}
		pMsg++;
		sLen++;
	}
	return( crc );
}
```

‘Cálculo del CRC

‘NOTA: Aunque el CRC es un INTEGER, se trabaja con LONG con el fin de poder tratar el bit de más peso de un entero (el de signo) sin problemas. BASIC no maneja números sin signo.

```
Function CRC16( Datos As String ) As Long
	Dim flag	As Long
	Dim crc	As Long
	Dim car	As Integer
	Dim bit		As Integer

	crc = &hFFFF&
	For car = 1 To Len( Datos )
		crc = crc Xor Asc(Mid$( Datos, car, 1 ))
		For bit = 0 To 7
			flag = crc And &h1&
			crc = crc \ 2&
			If flag = 1& Then crc = crc Xor &hA001&
		Next bit
	Next car

	CRC16 = crc
End Function
```

Ejemplo de comunicaciones Modbus

Seguidamente se muestra un ejemplo de comunicaciones Modbus en Visual Basic
»Comm» *es el nombre que se le ha dado al control MSCOMM.VBX insertado en el proyecto.*

```
Option Explicit
Private Sub Form_Load()
	Dim res As String		' Datos recibidos por el canal serie
	Dim datos	As String		' Datos a enviar por el canal serie
	Dim iVal	As Integer

	With Comm				'Parámetros de comunicaciones
		.InputMode = comInputModeBinary
		.CommPort = 1		'Port serie 1 ó 2
		.Settings = «9600,N,8,1»	'Parámetros de Comunicaciones
		.PortOpen = True		'Apertura del port
	End With

	'Pedir el valor de la dirección de memoria 2:
	'	1=NºDispositivo
	'	4=Código de Función
	'	2=Dirección de la palabra inicial a leer
	'	1=cantidad de palabras a leer
	res = Preguntar( 1, 4, 2, 1 )

	'iVal tiene el valor INTEGER de la dirección de memoria 1.
	iVal = BinToInt( res )

	'Ejemplo de escritura:
	'Modificar el dato que hay en la dirección de memoria 26,
	'asignando el valor 33.

	Const AL1 = 33
	Const DirW = 26
	datos = Chr$( AL1 \ 256 ) & Chr$( AL1 Mod 256 )
	res = Enviar( 1, DirW, 1, datos )

	Comm.PortOpen = False		'Cerrar las comunicaciones
	End
End Sub

'Ejemplo de lectura de N palabras del área de lectura o escritura.
'	nDIS = Número de Dispositivo
'	cod = Código de lectura: 3 ó 4
'	add = Palabra inicial a leer
'	cant = Cantidad de palabras a leer
'Retorna los datos recibidos vía serie
'
Function Preguntar (nDIS As Integer, cod As Integer, add As Integer, cant As Integer) As String
	Dim nBytes	As Integer		'Nº de bytes a recibir
	Dim hByte	As Integer		'Byte alto del CRC
	Dim lByte	As Integer		'Byte bajo del CRC
	Dim crc As Long		'Valor del CRC
	Dim env As String		'Cadena a enviar
	Dim TBytes()	As Byte		'Igual que «env»
	Dim b		As Integer		'Índice
```

```
	nbytes = 3 + (cant * 2) + 2			'Bytes a recibir
	env = Chr$( nDIS ) & Chr$( cod )			'cabecera
	env = env & Chr$(0) & Chr$( add )			'palabra inicial
	env = env & Chr$(0) & Chr$( cant )			'cantidad de palabras

	'Añadir el CRC al mensaje
	crc = CRC16( env )				'Calcular el CRC
	hByte = CInt(( crc And &HFF00& ) \ &H100& ) And &HFF&	'byte alto del CRC
	lByte = CInt( crc And &HFF& )			'byte bajo del CRC
	env = env & Chr$( lByte )
	env = env & Chr$( hByte )

	'Enviar el mensaje
	ReDim TBytes(1 To Len(env))			As Byte
	While b < Len( env )
		b = b + 1
		TBytes( b ) = AscB(Mid( env, b, 1 ))
	Wend
	Comm.Output = TBytes()

	'Esperar la respuesta
	While Comm.InBufferCount <> nbytes
		DoEvents
	Wend

	'Retornar la información recibida
	Erase TBytes
	b = 0
	env = «»
	TBytes() = Comm.Input
	While b < nbytes
		env = env & Chr$( TBytes( b ))
		b = b + 1
	Wend

	Preguntar = env
End Function
```

<u>'Cálculo del CRC</u>

'NOTA: Aunque el CRC es un INTEGER, se trabaja con LONG, con el fin de poder tratar el bit de 'más peso de un entero (el de signo) sin problemas. BASIC no maneja números sin signo.

```
Function CRC16( Datos As String ) As Long
	Dim flag	As Long
	Dim crc		As Long
	Dim car		As Integer
	Dim bit		As Integer

	crc = &hFFFF&
	For car = 1 To Len( Datos )
		crc = crc Xor Asc(Mid$( Datos, car, 1 ))
		For bit = 0 To 7
			flag = crc And &h1&
			crc = crc \ 2&
			If flag = 1& Then crc = crc Xor &hA001&
		Next bit
	Next car

	CRC16 = crc
End Function

'Enviar datos al Área de Escritura
'	nDIS = Número de dispositivo, add = Dirección inicial de escritura
'	cant = Cantidad de palabras a escribir, datos = Datos, en binario, a escribir
'Retorna los datos recibidos vía serie
Function Enviar(nDIS As Integer, add As Integer, cant As Integer, datos As String) As String
	Dim nbytes	As Integer	'Bytes a retornar el dispositivo
	Dim balto	As Integer	'Byte alto del CRC
	Dim bbajo	As Integer	'Byte bajo del CRC
	Dim crcAs Long		'Valor de CRC
	Dim env		As String	'Datos a enviar
	Dim TBytes()	As Byte	'Igual que «env»
	Dim b		As Integer	'Índice

	'Mensaje a enviar
	env = Chr$( nDIS ) & Chr$( 16 )		'cabecera
	env = env & Chr$(0) & Chr$( add )	'dirección inicial
	env = env & Chr$(0) & Chr$( cant )	'cantidad de palabras
	env = env & Chr$(cant * 2)	'nº de bytes
	env = env & datos
	nbytes = 8

	'Añadir el CRC al mensaje
	crc = CRC16(env)			'Calcular el CRC
	balto = CInt((( crc And &HFF00& ) \ &H100& ) And &HFF& )
	bbajo = CInt( crc And &HFF& )
	env = env & Chr$( bbajo )
	env = env & Chr$( balto )
```

```
    'Enviar los datos
    ReDim TBytes(1 To Len( env ))        As Byte
    While b < Len( env )
        b = b + 1
        TBytes( b ) = AscB(Mid( env, b, 1 ))
    Wend
    Comm.Output = TBytes()

    'Esperar la respuesta
    While Comm.InBufferCount <> nbytes: DoEvents: Wend

    'Retornar los datos recibidos
    Erase TBytes
    b = 0
    env = «»
    TBytes() = Comm.Input
    While b < nbytes
        env = env & Chr$(TBytes( b ))
        b = b + 1
    Wend

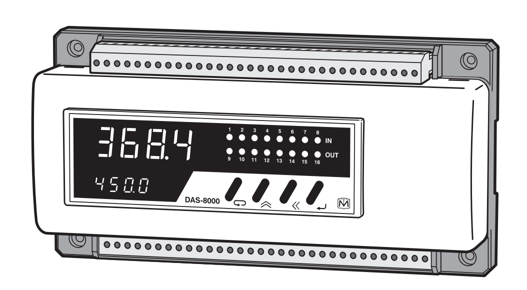
    Enviar = env
End Function

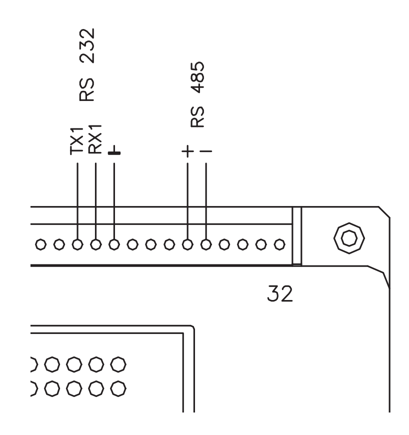
' Esta función convierte un valor 'Modbus' a un valor decimal
Function BinToInt( datos As String ) As Integer
    Dim sNum     As String
    Dim lNum     As Long
    Dim iRet     As Integer

    sNum = Mid$( datos, 4, 2 )
    lNum = 0
    lNum = lNum Or Asc(Mid$(sNum, 1, 1))     'byte alto
    lNum = lNum * &H100&
    lNum = lNum Or Asc(Mid$(sNum, 2, 1))     'byte bajo
    iRet = CInt(lNum And &H7FFF&)
    If (lNum And &H8000&) = &H8000& Then
        iRet = iRet Or &H8000
    End If
    BinToInt = iRet
End Function
```

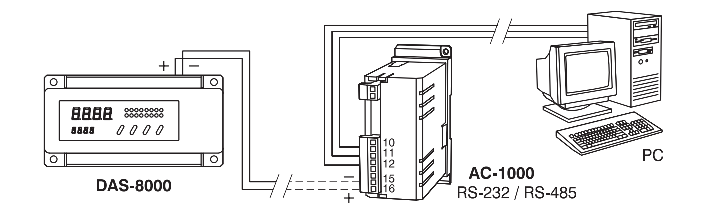
SALIDA RS-485 y RS-232

SERIE DAS-8000

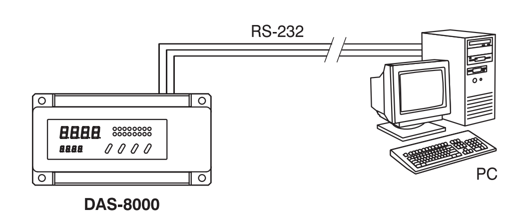




CONEXIONADO RS-485

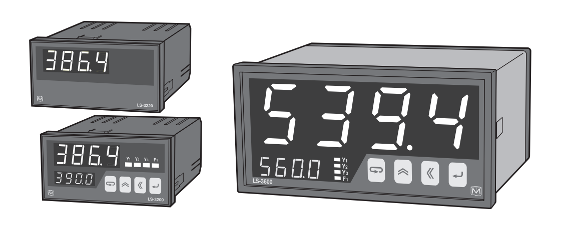


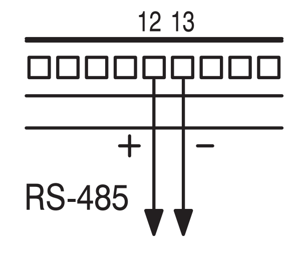
CONEXIONADO RS-232

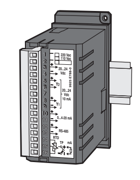


SALIDA RS-485

SERIE LS-3000





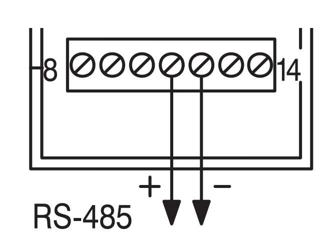




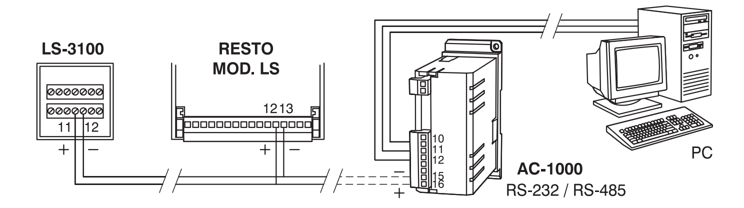


FORMATO LS-3100



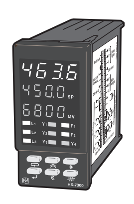


CONEXIONADO RS-485

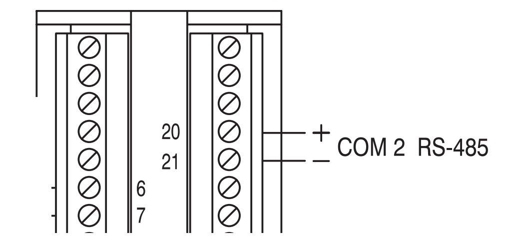


SALIDA RS-485

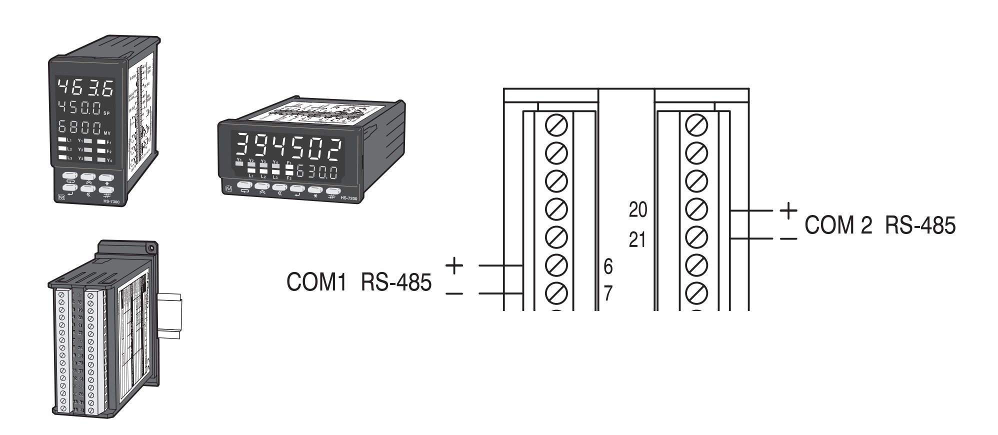
SERIE MS-5000



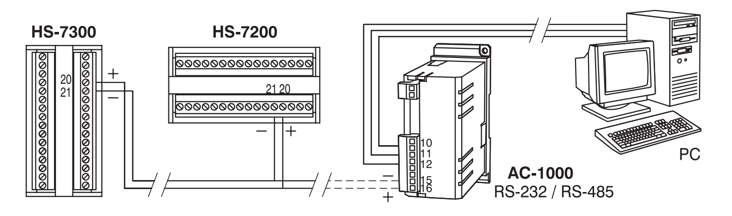




SERIE HS-7000



CONEXIONADO RS-485



APÉNDICE 1. - CÓDIGO HEXADECIMAL

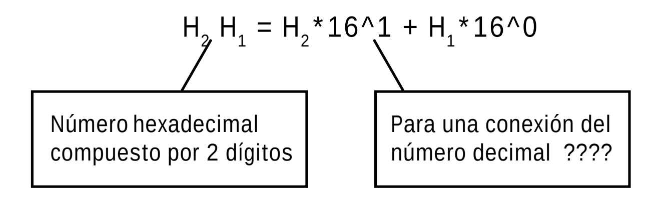
El Código Hexadecimal es la forma de contar usada normalmente en proceso de datos. De tal forma que todos los instrumentos actuales, basados en microprocesador utilizan un programa interno que usa este código, permitiéndoles manejar con gran facilidad todos los datos de entradas y salidas empleados internamente.

Este código está compuesto de 16 carácteres: los primeros 10 dígitos son numéricos al que les siguen las primeras 6 letras del alfabeto, presentando la siguiente sucesión:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Los primeros 10 carácteres coinciden con el código decimal usado habitualmente, seguidos de la letra A usada como 10, la B usada como 11, la C como 12, la D como 13, la E como 14, y la F como 15. A continuación, tal como se hace en decimal, después del 15, se comienza a contar de nuevo incrementando en 1 el dígito precedente, etc.

La fórmula por convertir el código Hexadecimal a Decimal para un valor compuesto de 2 dígitos es la siguiente:



Por ejemplo:

El valor Hexadecimal 1F correspondería a 1*16^1 + 15*16^0 que daría 16 +15 = 31 en Decimal.

APÉNDICE 2. LISTADO DE CODIGOS HEXADECIMALES

La tabla siguiente muestra los códigos hexadecimales de los primeros 255 dígitos decimales. Esta correspondencia también puede usarse para asignar direcciones a dispositivos en orden hexadecimal hasta 255, conectados a una red de comunicación.

DEC.	HEX.	DEC.	HEX.	DEC.	HEX.	DEC.	HEX.	DEC.	HEX.	DEC.	HEX.
1	1	44	2C	87	57	130	82	173	AD	216	D8
2	2	45	2D	88	58	131	83	174	AE	217	D9
3	3	46	2E	89	59	132	84	175	AF	218	DA
4	4	47	2F	90	5A	133	85	176	B0	219	DB
5	5	48	30	91	5B	134	86	177	B1	220	DC
6	6	49	31	92	5C	135	87	178	B2	221	DD
7	7	50	32	93	5D	136	88	179	B3	222	DE
8	8	51	33	94	5E	137	89	180	B4	223	DF
9	9	52	34	95	5F	138	8A	181	B5	224	E0
10	A	53	35	96	60	139	8B	182	B6	225	E1
11	B	54	36	97	61	140	8C	183	B7	226	E2
12	C	55	37	98	62	141	8D	184	B8	227	E3
13	D	56	38	99	63	142	8E	185	B9	228	E4
14	E	57	39	100	64	143	8F	186	BA	229	E5
15	F	58	3A	101	65	144	90	187	BB	230	E6
16	10	59	3B	102	66	145	91	188	BC	231	E7
17	11	60	3C	103	67	146	92	189	BD	232	E8
18	12	61	3D	104	68	147	93	190	BE	233	E9
19	13	62	3E	105	69	148	94	191	BF	234	EA
20	14	63	3F	106	6A	149	95	192	C0	235	EB
21	15	64	40	107	6B	150	96	193	C1	236	EC
22	16	65	41	108	6C	151	97	194	C2	237	ED
23	17	66	42	109	6D	152	98	195	C3	238	EE
24	18	67	43	110	6E	153	99	196	C4	239	EF
25	19	68	44	111	6F	154	9A	197	C5	240	F0
26	1A	69	45	112	70	155	9B	198	C6	241	F1
27	1B	70	46	113	71	156	9C	199	C7	242	F2
28	1C	71	47	114	72	157	9D	200	C8	243	F3
29	1D	72	48	115	73	158	9E	201	C9	244	F4
30	1E	73	49	116	74	159	9F	202	CA	245	F5
31	1F	74	4A	117	75	160	A0	203	CB	246	F6
32	20	75	4B	118	76	161	A1	204	CC	247	F7
33	21	76	4C	119	77	162	A2	205	CD	248	F8
34	22	77	4D	120	78	163	A3	206	CE	249	F9
35	23	78	4E	121	79	164	A4	207	CF	250	FA
36	24	79	4F	122	7A	165	A5	208	D0	251	FB
37	25	80	50	123	7B	166	A6	209	D1	252	FC
38	26	81	51	124	7C	167	A7	210	D2	253	FD
39	27	82	52	125	7D	168	A8	211	D3	254	FE
40	28	83	53	126	7E	169	A9	212	D4	255	FF
41	29	84	54	127	7F	170	AA	213	D5		
42	2A	85	55	128	80	171	AB	214	D6		
43	2B	86	56	129	81	172	AC	215	D7		



DESIN INSTRUMENTS S.A.

Av. Frederic Rahola, 49 - 08032 BARCELONA (España)
Tel. (+34) 93 358 6011* - Fax (+34) 93 357 6850
e-mail:desin@desin.com - http://www.desin.com